Auction House Price Analysis

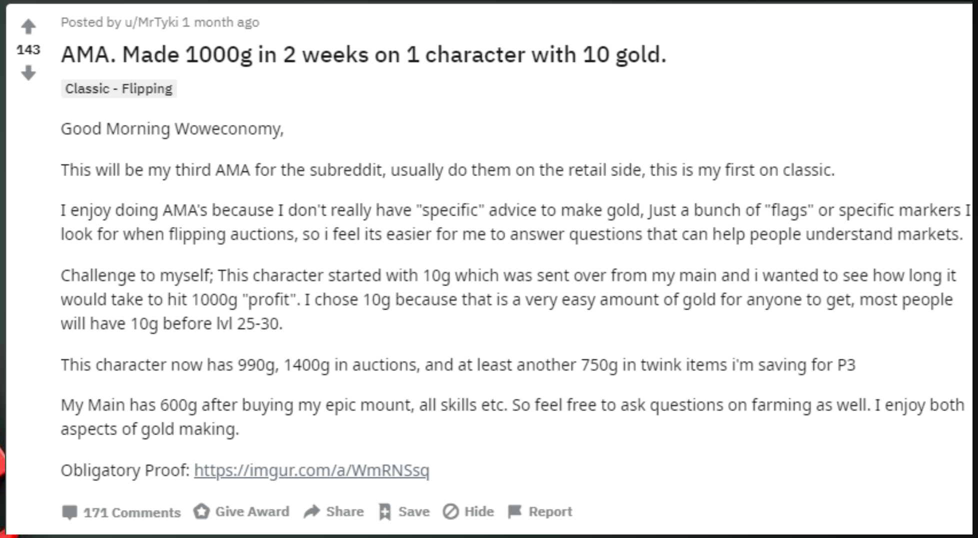
Nikolai Alexander – CSCI4502
nial3328@colorado.edu

The Auction House

- Faction-Wide Hub for Player-To-Player Trading
- Dynamic, Highly Volatile Economy
- Player Driven Competitive Pricing
- Prices Affected by Demand/Popularity, Level, Relevance

The Auction House

- Players Exploit Behavior for Profit
- No need for collecting items in world
- Generates gold faster than any other method of income



Posted by u/MrTyki 1 month ago

143

AMA. Made 1000g in 2 weeks on 1 character with 10 gold.

Classic - Flipping

Good Morning Woweconomy,

This will be my third AMA for the subreddit, usually do them on the retail side, this is my first on classic.

I enjoy doing AMA's because I don't really have "specific" advice to make gold, Just a bunch of "flags" or specific markers I look for when flipping auctions, so i feel its easier for me to answer questions that can help people understand markets.

Challenge to myself; This character started with 10g which was sent over from my main and i wanted to see how long it would take to hit 1000g "profit". I chose 10g because that is a very easy amount of gold for anyone to get, most people will have 10g before lvl 25-30.

This character now has 990g, 1400g in auctions, and at least another 750g in twink items i'm saving for P3

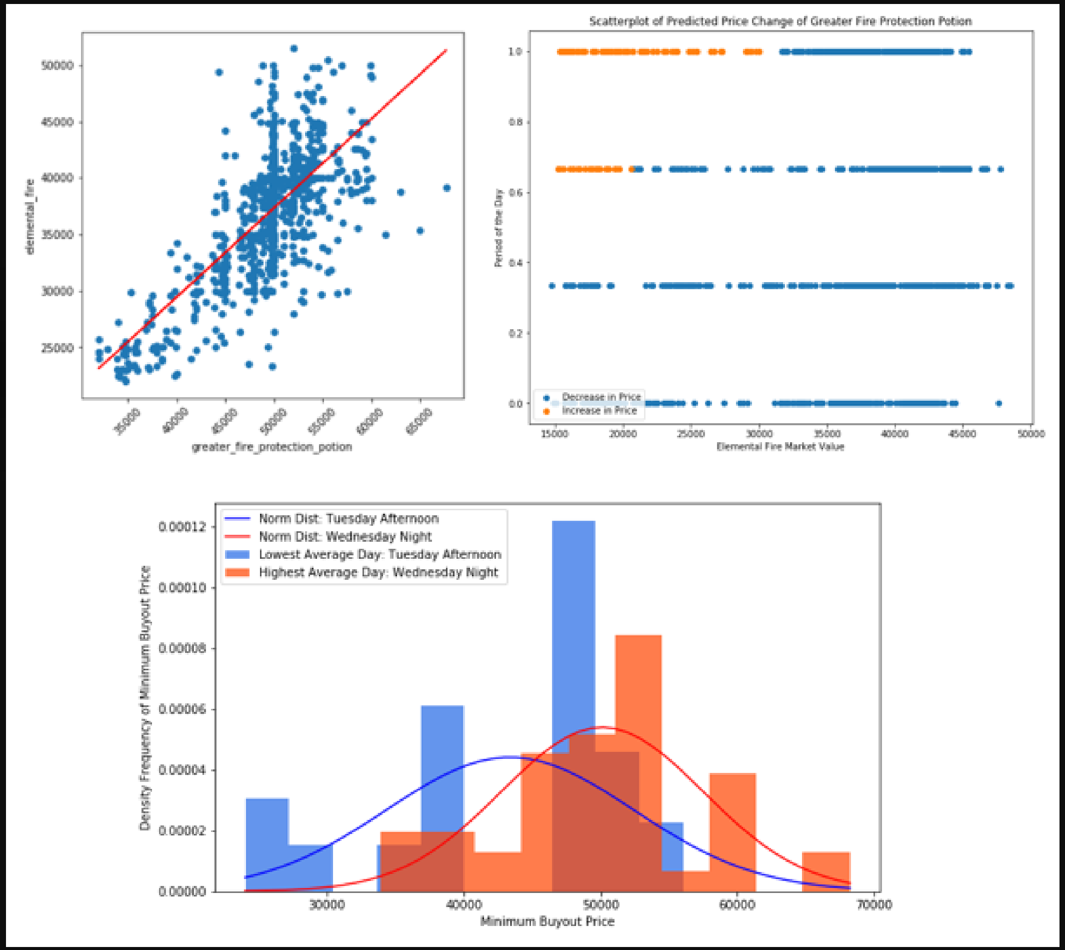
My Main has 600g after buying my epic mount, all skills etc. So feel free to ask questions on farming as well. I enjoy both aspects of gold making.

Obligatory Proof: https://imgur.com/a/WmRNSsq

171 Comments Give Award Share Save Hide Report

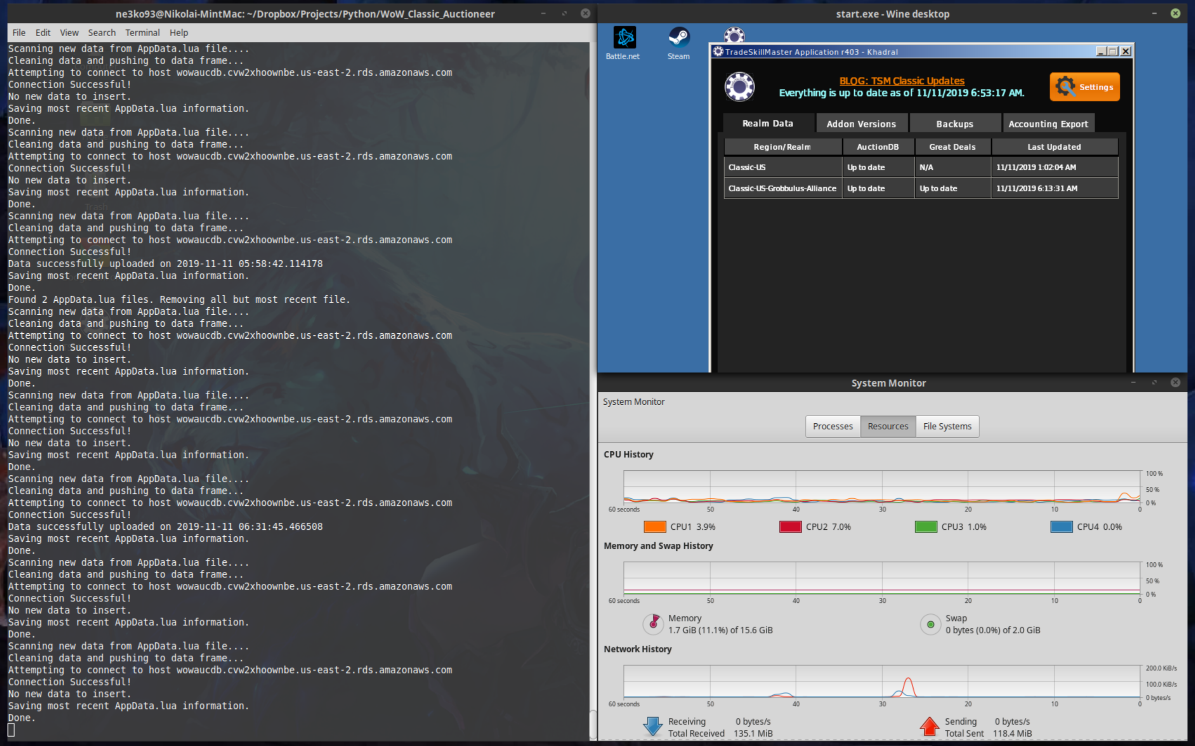
Analyzing Price Behavior

- Simple Linear Regression
  - Correlations between Craftables & Materials
- Confidence Interval Estimation
  - Difference in mean price between Time Intervals
- K-Means Clustering
  - Predicting Price Change Behavior



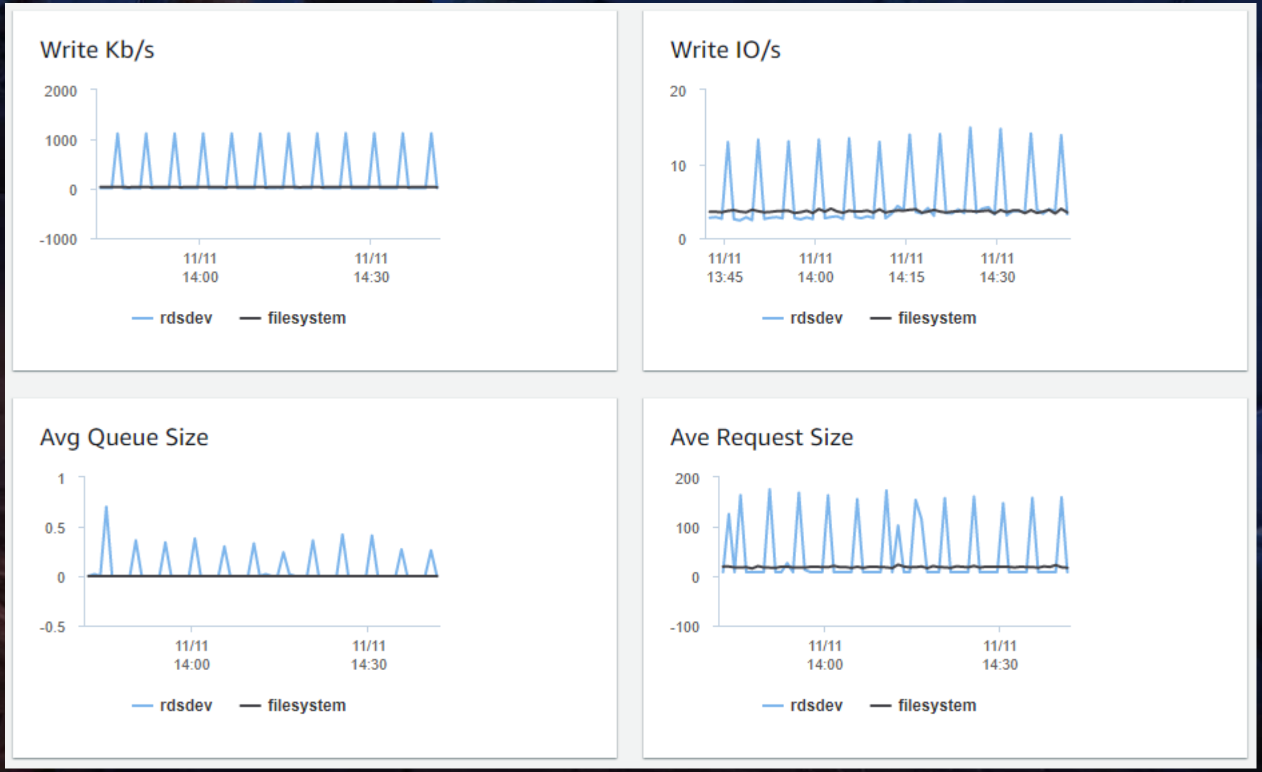
Data Collection

- Script perpetually running on Linux server
- Scans TradeSkillMaster data file every second – AppData.lua
- If update to file, parses and preprocesses data (JSON format)
- Pushes processed data to AWS server



AWS Database

- 5966 Items
- ~Hourly scans dating back to October 7, 2019
- Attributes: itemid, marketvalue, minbuyout, historical, numauctions, scantime
    - Predicted variable based off minbuyout



Simple Linear Regression - Algorithm

- Inputs
  - Item ID of Craftables,
  - Item Name of Craftable
  - List of Item IDs of Materials
  - List of Names of Materials
- Output
  - Summary of Linear Correlations Ordered by Pearson Correlation Coefficient
  - Plots of Pairwise SLR Relationships
- Steps
  1. Extract scantimes & minimum buyout prices for all items
  2. Merge Data Sets
  3. Produce Pearson Correlation Coefficient Matrix
     1. pandas.dataframe.corr()
  4. Model SLR Line for each pairwise relationship
     1. Estimate Slope β & y-intercept α
     2. Calculate t-statistic
     3. Calculate p-value
        1. scipy.stats.t.cdf()
     4. Calculate Coefficient of Determination R^2

Simple Linear Regression - Mathematics

- Pearson Correlation Coefficient

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\,\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

- Slope β

$$\hat{\beta} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

- Y-Intercept α

$$\hat{a} = \bar{y} - \hat{\beta}\bar{a}$$

- T-statistic

$$t = \frac{\hat{\beta}}{SE(\hat{\beta})}$$

- Standard Error

$$SE(\hat{\beta}) = \sqrt{\frac{1}{n-2}\frac{\sum_{i=1}^{n}(y_i - \hat{y})^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}}$$

- P-value

$$p = \begin{cases} 1 - \Phi(t) \;\; if\ \hat{\beta} \le 0 \\ \Phi(t)\ if\ \hat{\beta} > 0 \end{cases}$$

- Coefficient of Determination

$$R^2 = 1 - \frac{SSR}{SST}$$

- Residual Sum of Squares

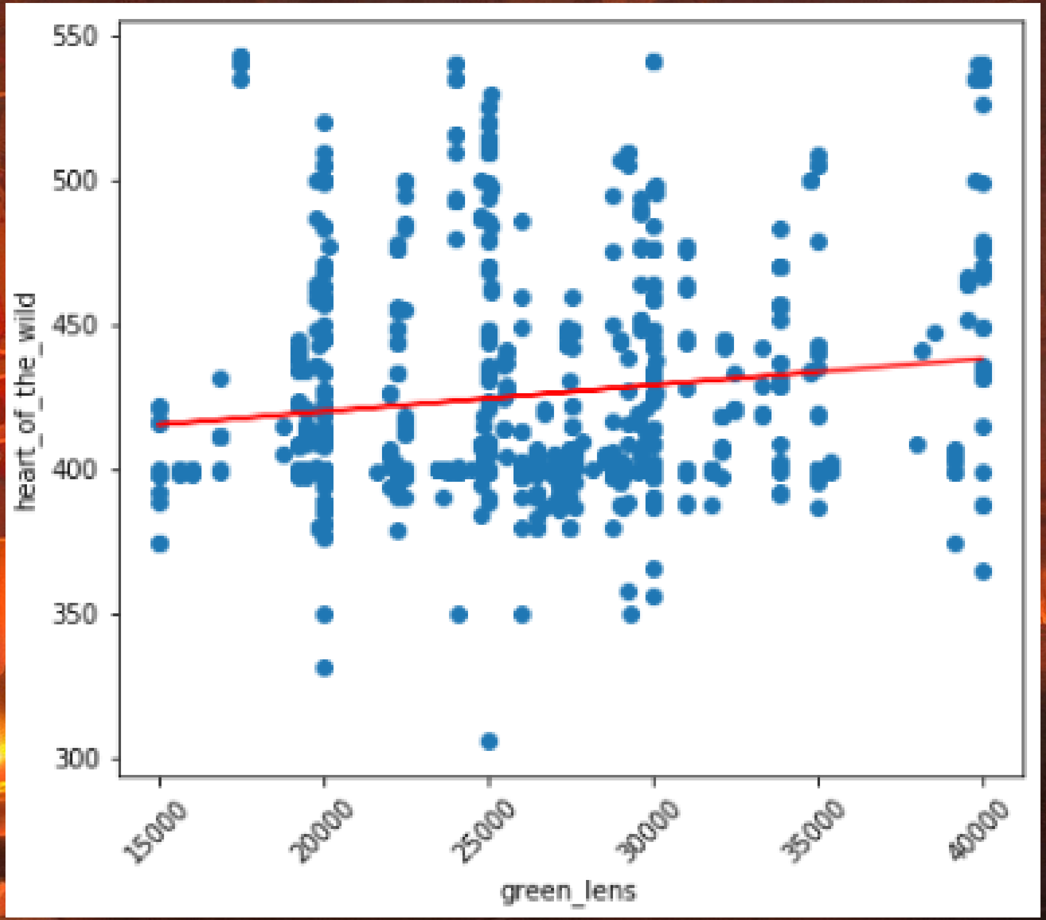
$$SSR = \sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2$$

- Total Sum of Squares

$$SST = \sum_{i=1}^{n}(y_i - \bar{y})^2$$

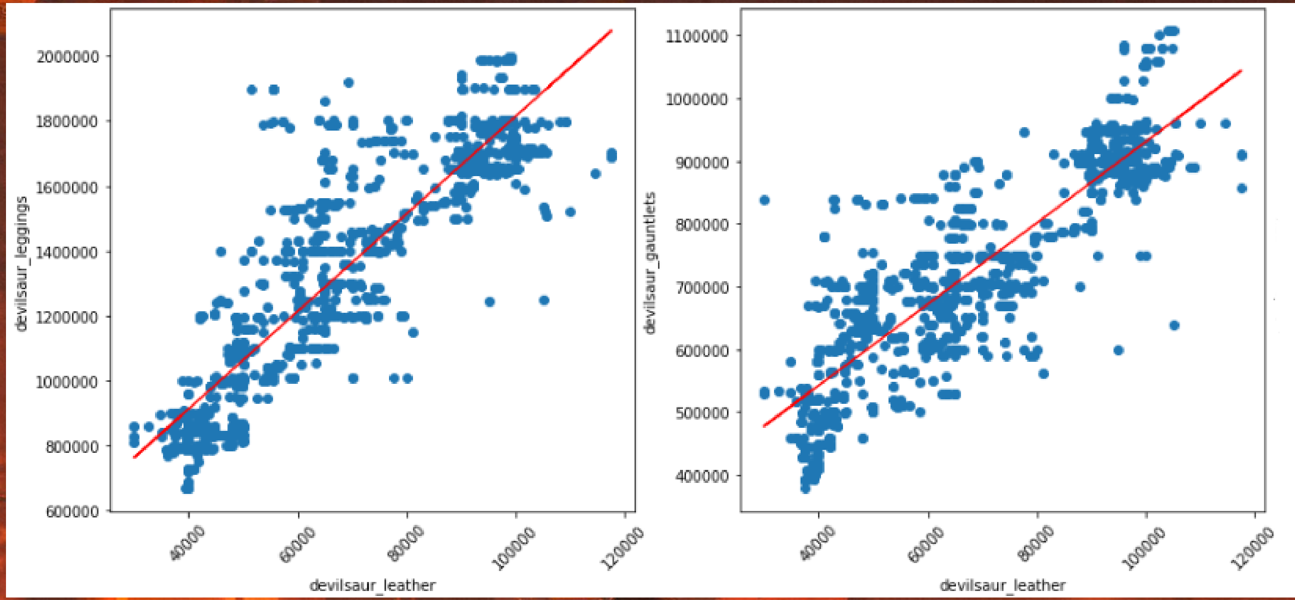
Simple Linear Regression – BiS Results

- Generally Quite Poor Performance
- Green Lens vs Heart of the Wild
  - $p = 2.84e^{-5}$
  - $r_{xy} = 0.137$
  - $R^2 = 0.0188$



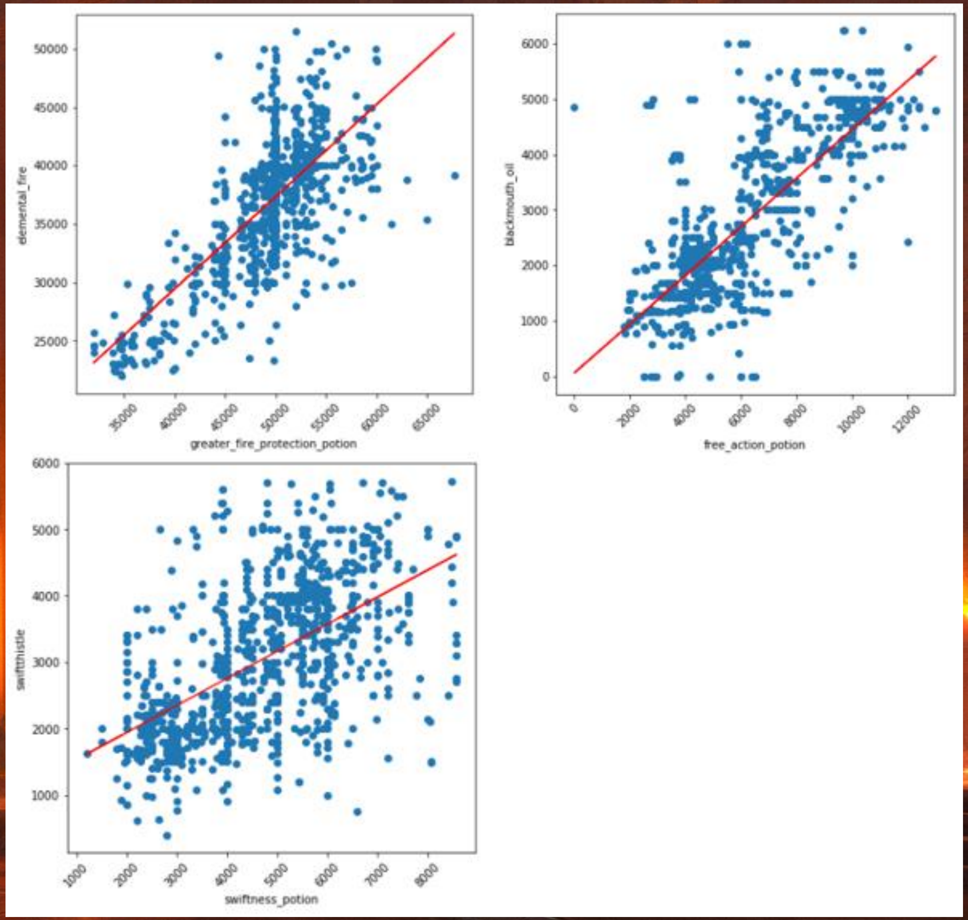
Simple Linear Regression – Devilsaur Leather

- Contrary to other BiS, performed well
- Devilsaur Leggings vs Devilsaur Leather
  - $p = 0.00$
  - $r_{xy} = 0.81$
  - $R^2 = 0.67$
- Devilsaur Gauntlets vs Devilsaur Leather
  - $p = 0.00$
  - $r_{xy} = 0.828$
  - $R^2 = 0.68$



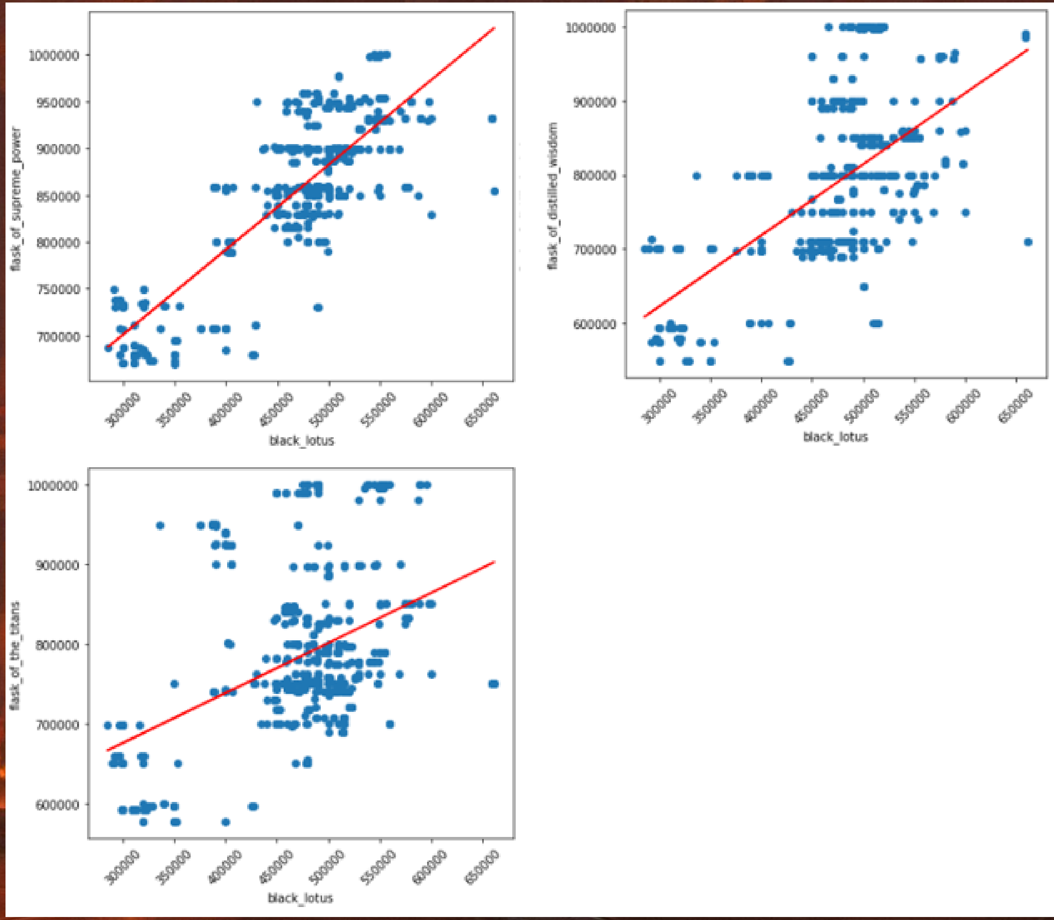
Simple Linear Regression – Consumables

- Consistently performed better than BiS
- Greater Fire Protection Potion vs Elemental Fire
  - $p = 0.00, r_{xy} = 0.72, R^2 = 0.52$
- Free Action Potion vs Blackmouth Oil
  - $p = 0.00, r_{xy} = 0.78, R^2 = 0.608$
- Swiftness Potion vs Swiftthistle
  - $p = 0.00, r_{xy} = 0.57, R^2 = 0.327$



Simple Linear Regression – Flasks

- All share same main ingredient – Black Lotus
- Correlation proportional to player population
- Flask of Supreme Power vs Black Lotus
  - $p = 0.00, r_{xy} = 0.81, R^2 = 0.657$
- Flask of Distilled Wisdom vs Black Lotus
  - $p = 0.00, r_{xy} = 0.607, R^2 = 0.369$
- Flask of the Titans vs Black Lotus
  - $p = 0.00, r_{xy} = 0.431, R^2 = 0.186$



Time-Interval Analysis – Methodology

- Split minimum buyout prices into subsets based on time interval
  - Day of the Week, Hour of the Day, Part of the Day (Morning, Afternoon, Evening, Night)
- Compare Best Performing Interval and Worst Performing Interval
  - Performance measured by sample mean
- Significance measured by p-value within significance level $\alpha = 0.05$

- T-statistic for difference in means

$$t = \frac{\mu_1 - \mu_2}{\sqrt{\sigma_1^2/n_1}\sqrt{\sigma_2^2/n_2}}$$

- Degrees of Freedom

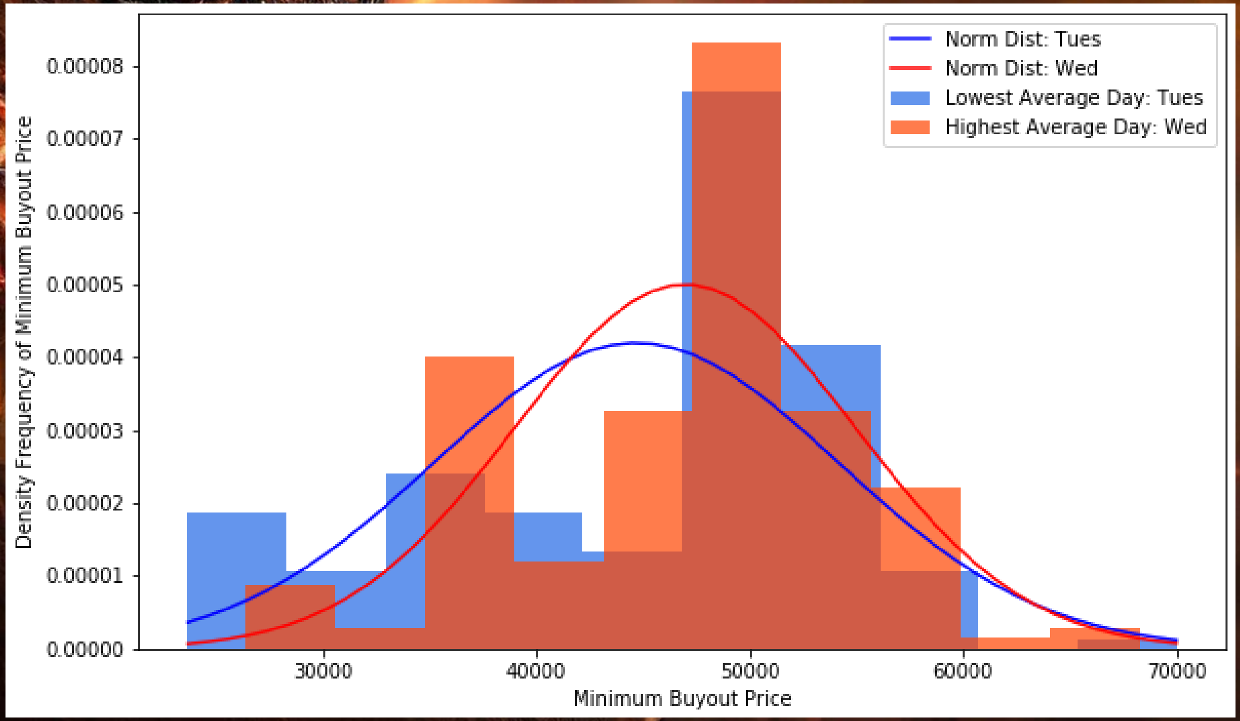
$$df = \frac{\left(\sigma_1^2/n_1 + \sigma_2^2/n_2\right)^2}{\left(\sigma_1^2/n_1\right)^2/(n_1 - 1) + \left(\sigma_2^2/n_2\right)^2/(n_2 - 1)}$$

- p-value

$$p = 1 - \phi(t)$$

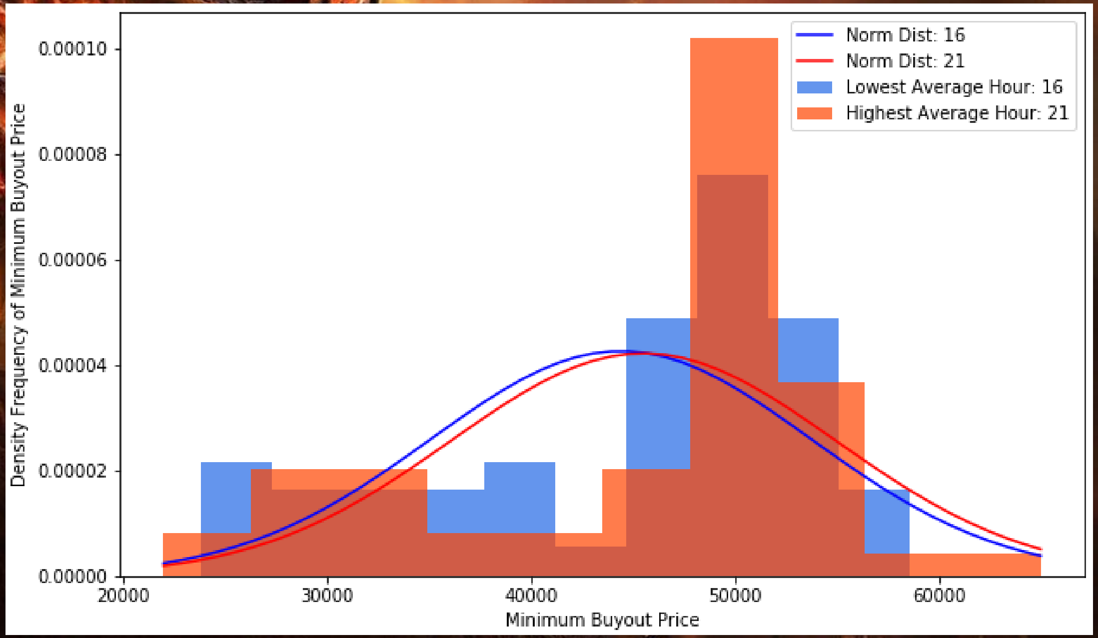
Time-Interval Analysis – Weekday Intervals

- Best Performing Day – Wednesday
  - $\mu = 4g\ 69s\ 84c,\ \sigma = 79s\ 81c$
- Worst Performing Day – Tuesday
  - $\mu = 4g\ 46s\ 94c,\ \sigma = 95s\ 06c$
- P-value: $p = 0.0102$



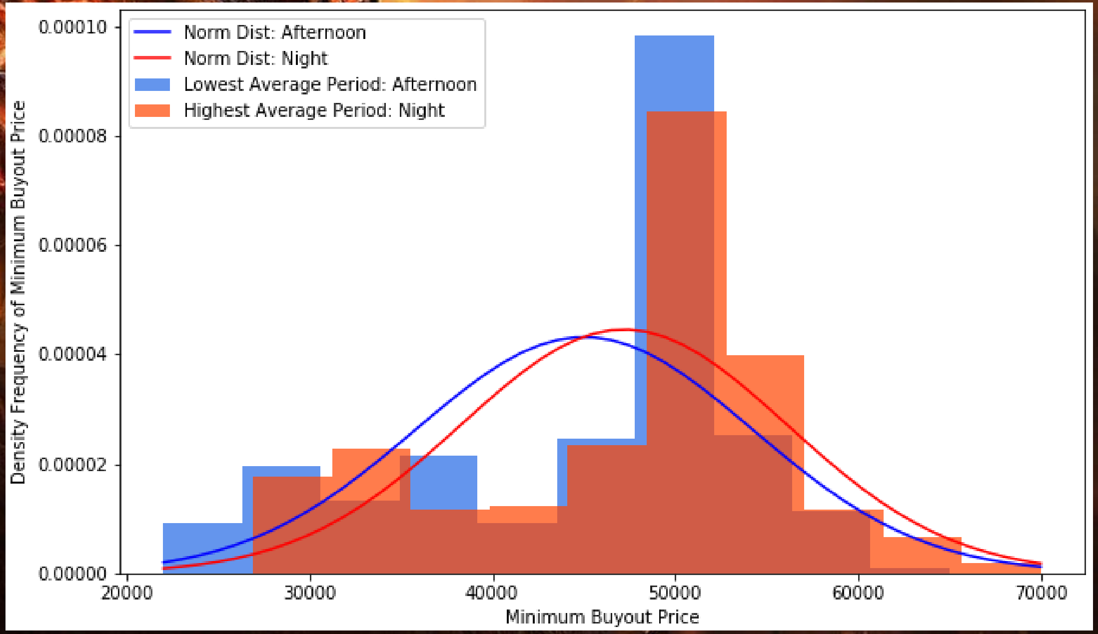
Time-Interval Analysis – Hourly Intervals

- Best Performing Hour – 9pm-10pm
  - $\mu = 4g\ 84s\ 10c, \sigma = 88s\ 36c$
- Worst Performing Day – Tuesday
  - $\mu = 4g\ 42s\ 76c, \sigma = 94s\ 46c$
- P-value: $p = 0.2712$



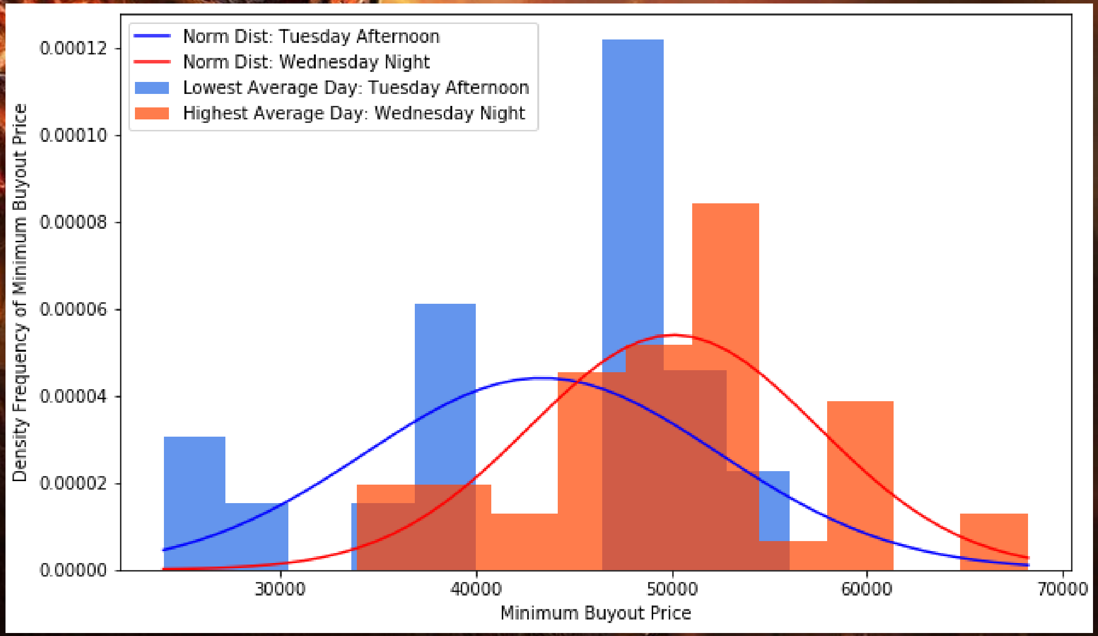
Time-Interval Analysis – Part of Day Intervals

- Best Performing PoD – Night (9pm-5am)
  - $\mu = 4g\ 71s\ 98c, \sigma = 89s\ 58c$
- Worst Performing Day – Afternoon (12pm-5pm)
  - $\mu = 4g\ 50s\ 17c, \sigma = 92s\ 44c$
- P-value: $p = 0.00137$



Time-Interval Analysis – Theoretical Best vs Theoretical Worst

- Wednesday Night (Theoretical Best)
  - $\mu = 5g\ 01s\ 26c, \sigma = 74s\ 02c$
- Tuesday Afternoon (Theoretical Worst)
  - $\mu = 4g\ 33s\ 79c, \sigma = 90s\ 60c$
- P-value: $p = 0.000164$



K-Means Clustering - Algorithm

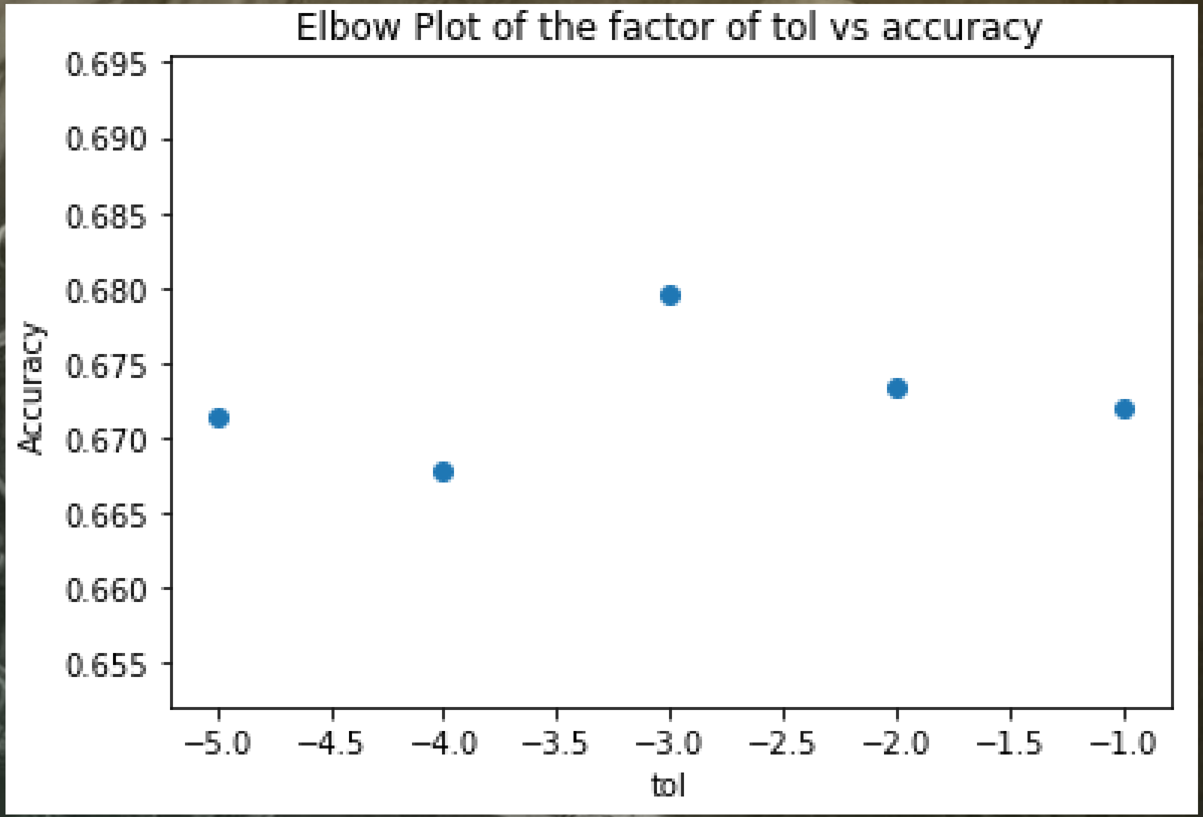
- Input
  - Matrix of Predictor Variables
  - Real Class Values
  - K value (# of classes)
  - Stopping Condition
    1. Number of iterations
    2. Convergence of Centroids
  - Distance Metric
    - Euclidean, Manhattan, Infinite, Cosine, Lr
    - r for Lr distance
- Output
  - Matrix of Centroids for each Cluster
  - Predicted Classes
  - Accuracy of Prediction
  - Confusion Matrix

K-Means Clustering - Algorithm

- Initialization
  - Centroids
    - First centroid picked as random point
    - Following centroids picked as furthest distance from previous centroids
  - Initial Predicted Classes – Centroid of minimum distance from point
- Iteration
  - Assign class for each instance by minimum distance from centroids
  - Update centroids to be center of each new cluster
- Convergence
  1. Reach assigned number of iterations
  2. Centroids converge to central point

K-Means Clustering - Tuning

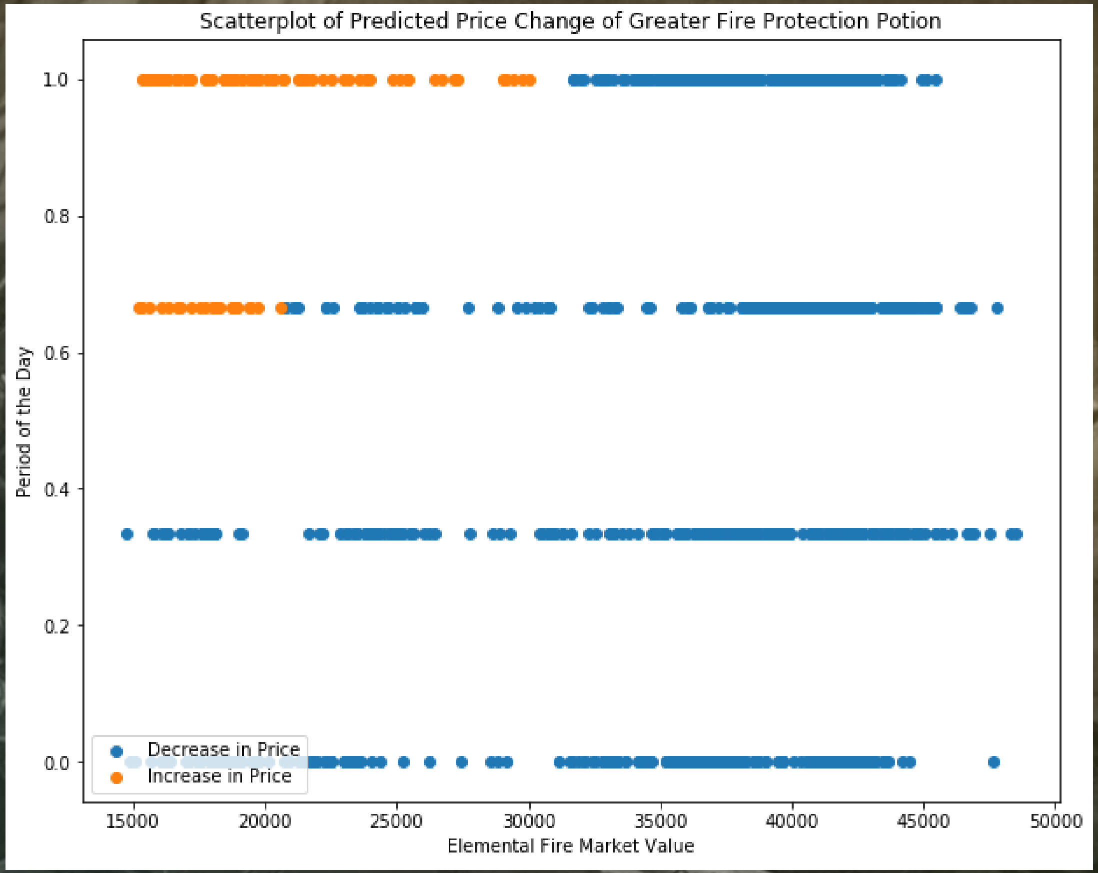
- Best Attributes (significance level $\alpha = 0.05$)
  - Part of the Day
    - Pearson Correlation: 0.024
    - P-value: 0.00789
  - Elemental Fire Market Value
    - Pearson Correlation: -0.015
    - P-value: -0.0104
- Optimal Distance Metric
  - Cosine Distance (68% Accuracy)
  - All other distances around (58% Accuracy)
- Optimal Stopping Condition
  - Convergence of Centroids
  - Difference of 0.001



K-Means Clustering - Results

- Accuracy: 70.17%
- Confusion Matrix
  - 91% Bias towards Decreasing / No Change

	Decrease/ No Change	Increase
Decrease/ No Change	745	76
Increase	250	22



Conclusion

- Predicting price trends with Confidence Interval Estimation of worst/best performing time-intervals gave the best results.
  - Tuesday Afternoon is the best time to buy Greater Fire Protection Potion, while Wednesday Night is the best time to sell.
    - Correlates with weekly server reset and raiding schedules
- K-means cluster is the worst performing method of prediction for this situation
  - Only 70% accuracy
  - Too many variables and too much randomness in price to predict specific price changes